準同型暗号と整数及び整数多項式の近似GCD (数式処理研究の新たな発展)

AUTHOR(S):

長坂, 耕作

CITATION:

長坂, 耕作. 準同型暗号と整数及び整数多項式の近似GCD (数式処理研究の新たな発展). 数理解析研究所講究録 2011, 1759: 115-123

ISSUE DATE:

2011-09

URL:

http://hdl.handle.net/2433/171328

数理解析研究所講究録
第1759巻 2011年 115-123

準同型暗号と整数及び整数多項式の近似GCD
Homomorphic Encryption and Approximate GCD of Integers and Polynomials over Integers *

長坂耕作
KOSAKU NAGASAKA†
神戸大学人間発達環境学研究科
GRADUATE SCHOOL OF HUMAN DEVELOPMENT AND ENVIRONMENT, KOBE UNIVERSITY

Abstract

Communications of the ACM の記事に基づいて，導入的な準同型暗号と整数の近似 GCD の関係を紹介した上で，整数の近似 GCD と整数係数多項式の近似 GCD との関連，加えて今後の展望について取り上げる。特に，格子算法を用いた整数係数多項式の近似 GCD 算法を，整数の近似 GCD 計算に拡張する試みについて詳しく述べ，桁上がりと桁下がりの問題から整数係数多項式用のアルゴリズムを整数に適用できないことを克服するアイデアについて報告する。

Abstract

We briefly review the article published in Communications of the ACM, about homomorphic encryption and approximate GCD of integers, and a known algorithm for approximate GCD of polynomials over integers. Extending algorithms for polynomials over integers to integers by mapping the variable to the base number is not easy since the integer arithmetic causes carry and bollow digits while the polynomial arithmetic does not have this property. In this preliminary report, we introduce a way to overcome this problem.

1 準同型暗号と整数の近似GCD

Gentry による Communications of the ACM の記事 [3] において，準同型暗号と整数の近似 GCD との関係が取り上げられている。ここでは，本報告で取り上げる内容に必要となる主要な部分について，記事に基づき簡単に紹介する。

1.1 準同型暗号とは

そもそも準同型暗号が求められる理由は複雑なものでなく，データをクラウドのような第三者に預けるだけでなく，そのクラウド上で計算 (データ処理) をしたいという自然な理由による。例えば，暗号化されたデータを処理したいが，1) 復号して平文を求めデータ処理，2) 結果を改めて暗号化して保存，という手順で処理を行うということは，平文を求めないとデータ処理が不可能であることを意味する。これは，第三者にデータの中身を秘匿したまま，データ処理の委託をすることが困難であることを意味する。これを

*本研究の一部は科研費（22700011）の支援で行われている。
†nagasaka@main.h.kobe-u.ac.jp

可能とするのが準同型暗号 (Homomorphic Encryption) であり，復号せずに平文を処理した結果の暗号文を得られる暗号方式をいう。特に全ての論理演算が可能な準同型暗号を「Fully Homomorphic Encryption Scheme」という。Gentry らによる論文から数学的な定義を引用すると次のようになる。

定義 1 (Correct Homomorphic Decryption)
The scheme $\mathcal{E} = (KeyGen, Encrypt, Decrypt, Evaluate)$ *is correct for a given t-input circuit* C *if, for any key-pair* (sk, pk) *output by* $KeyGen(\lambda)$*, any* t *plaintext bits* $m_1, \ldots, m_t$*, and any ciphertexts* $\vec{c} = \langle c_1, \ldots, c_t \rangle$ *with* $c_i \leftarrow Encrypt_{\mathcal{E}}(pk, m_i)$*, it is the case that:*

$$Decrypt(sk, Evaluate(pk, C, \vec{c})) = C(m_1, \ldots, m_t).$$

◁

定義 2 (Homomorphic Encryption)
The scheme $\mathcal{E} = (KeyGen, Encrypt, Decrypt, Evaluate)$ *is homomorphic for a class* $\mathcal{C}$ *of circuits if it is correct for all circuits* $C \in \mathcal{C}$*.* $\mathcal{E}$ *is fully homomorphic if it is correct for all boolean circuits.* ◁

1.2 準同型暗号の実際と近似 GCD との関係

簡易な準同型暗号として，記事や論文では「A Somewhat Homomorphic Encryption Scheme」というのが導入されており，それをここでも紹介しておく。なお，ここでは簡単のため対称鍵暗号の場合を取り上げるが，非対称鍵暗号の場合も同様に可能であるので，必要に応じて参考文献に記載の記事や論文を参照して欲しい。また，平文として取り得るのは $\{0,1\}$ となっているが，どのようなデータ処理もこれで十分であることに注意されたい。

安全度の指標 (Security Parameter)
λ を Security Parameter とし，それに依存する形で，$N = \lambda$, $P = \lambda^2$, $Q = \lambda^5$ とおく。

秘密鍵の生成 KeyGen(λ)
P ビットの奇整数 p をランダムに生成することで行う。

暗号化 Encrypt(p, m)
平文 $m \in \{0,1\}$ の秘密鍵 p による暗号化は，$m' \equiv m \pmod 2$ を満たす N ビットの整数 m' と Q ビットの整数 q をランダムに生成し，暗号文 c を $m' + pq$ とすることで行う。

復号 Decrypt(p, c)
暗号文 c の秘密鍵 p での復号は，$(c \bmod p) \bmod 2$ で行う。

この暗号方式は，下記のように簡単に確認できるが，暗号文の加減乗算は，演算回数が少なければ準同型であることがわかる。ただし，演算回数が多い場合に，$m'_1 \pm m'_2$ や $m'_1 m'_2$ などが秘密鍵である p よりも大きくなると正しい平文を復号することが出来なくなる。

$$c_1 \pm c_2 = (m'_1 + pq_1) \pm (m'_2 + pq_2) = m'_1 \pm m'_2 + p(q_1 \pm q_2),$$
$$c_1 \times c_2 = (m'_1 + pq_1)(m'_2 + pq_2) = m'_1 m'_2 + pq'$$

Gentry の記事や論文では，この制約 (演算回数が少なくてはならない) をなくすことで，完全な準同型暗号を作り出している。その考え方はシンプルであり，演算回数が増えたら，ノイズ (m'_1 や m'_2 の大部分は平文を隠蔽するためのノイズ) をリフレッシュすることで，秘密鍵 p よりも大きくならないようにするだけである。このリフレッシュを，平文に復号せずに行うため，新しい暗号化を行ってから前の暗号化を復号する

という手法で実現している。このような仕組みが可能なものを，Gentry は「Bootstrappable」と定義している (詳細に付いては，Gentry の記事や論文を参照されたい)。

この暗号方式の攻撃方法の一つとして，整数の近似 GCD を求めるものがある。攻撃に際して必要となる情報は，同一の秘密鍵 p で暗号化された暗号文を 2 つ以上 (例えば，c_1 と c_2 など) であり，入手できる暗号文が多いほど攻撃はし易くなる。この攻撃方法では，得られた複数の暗号文から秘密鍵を求めようとする。つまり，$c_1 = m_1' + pq_1$ と $c_2 = m_2' + pq_2$ から p が求められるかという問題であり，c_1 と c_2 の近似 GCD に帰着される (m_1' と m_2' を誤差と見做して，共通因子 p を取り出す計算)。Gentry らの記事や論文では，既知の方法に対して安全なパラメータ設定を行うことで，この脆弱性を回避している。本報告は，これらの事実に触発され，整数係数多項式の近似 GCD 算法により整数の近似 GCD を求めることが可能であるか，について試みた結果の速報である。

2 整数係数多項式の近似 GCD

整数係数多項式の近似 GCD 自体は，長年多くの研究者により研究が進められ，係数の摂動を実数上や複素数上で許容すれば，かなりの精度で計算が可能となっているが，摂動を整数上に限る問題に付いては余り研究が進んでおらず，確認できるのは参考文献に記した 2 件 [2, 5] のみである。Gathen らの論文 [2] では，Howgrave-Graham による整数の近似 GCD を，整数係数多項式の近似 GCD へ拡張しており，実用的でないものの，適用範囲の解析も行われている。ここでは，数少ない既知の成果として，著者が国際研究集会のポスター・セッションで発表したもの [5] を紹介する。まず，本報告で扱う整数係数多項式の近似 GCD の定義は次の通りである。

定義 3 (Approximate Polynomial GCD Over Integers)
Let $f(\vec{x})$ and $g(\vec{x})$ be polynomials in variables $\vec{x} = x_1, \ldots, x_\ell$ over $\mathbb{Z}$, and let ε be a small positive integer. If $f(\vec{x})$ and $g(\vec{x})$ satisfy

$$f(\vec{x}) = t(\vec{x})h(\vec{x}) + \delta_f(\vec{x}),\ \ g(\vec{x}) = s(\vec{x})h(\vec{x}) + \delta_g(\vec{x}),\ \ \varepsilon = \max\{\|\delta_f\|, \|\delta_g\|\},$$

*for some polynomials $\delta_f, \delta_g \in \mathbb{Z}[\vec{x}]$, then we say that the above polynomial $h(\vec{x})$ is an **approximate GCD over integers**. We also say that $t(\vec{x})$ and $s(\vec{x})$ are **approximate cofactors over integers**, and we say that their **tolerance** is ε. ($\|p\|$ denotes a suitable norm of polynomial $p(\vec{x})$.)* ◁

例 1 (2 変数多項式の近似 GCD)
次の 2 つの多項式 $f(x_1, x_2)$ と $g(x_1, x_2)$ の整数上の近似 GCD について取り上げる。

$$\begin{aligned} f(x_1,x_2) &= 89x_1^2x_2^2 - 87x_1x_2^2 - 136x_2^2 + 15x_1^2x_2 + 132x_1x_2 + 119x_2 - 42x_1^2 + 166x_1 + 139, \\ g(x_1,x_2) &= 56x_1^2x_2^2 - 45x_1x_2^2 - 98x_2^2 - 13x_1^2x_2 + 46x_1x_2 + 225x_2 - 12x_1^2 + 80x_1 - 112. \end{aligned}$$

近似 GCD は一意でないが，その 1 つとして次のように $(5x_1x_2 - 9x_2 - 3x_1 + 14)$ を求めることができる。なお，下線部は摂動の結果で変化した桁を表している。

$$\begin{aligned} f(x_1,x_2) &\approx (18x_1x_2 + 15x_2 + 14x_1 + 10)(5x_1x_2 - 9x_2 - 3x_1 + 14) \\ &= \underline{90}x_1^2x_2^2 - 87x_1x_2^2 - 13\underline{5}x_2^2 + 1\underline{6}x_1^2x_2 + 13\underline{1}x_1x_2 + 1\underline{20}x_2 - 42x_1^2 + 166x_1 + 1\underline{40}, \\ g(x_1,x_2) &\approx (11x_1x_2 + 11x_2 + 4x_1 - 8)(5x_1x_2 - 9x_2 - 3x_1 + 14) \\ &= 5\underline{5}x_1^2x_2^2 - 4\underline{4}x_1x_2^2 - 9\underline{9}x_2^2 - 13x_1^2x_2 + 4\underline{5}x_1x_2 + 22\underline{6}x_2 - 12x_1^2 + 80x_1 - 112. \end{aligned}$$

◁

2.1 整数係数多項式の近似 GCD 算法

近似 GCD の計算では，次の部分集結式写像 $Syl_r(f,g)$ が重要である。

$$Syl_r(f,g): \begin{array}{ccc} \mathcal{P}_{m-r-1} \times \mathcal{P}_{n-r-1} & \rightarrow & \mathcal{P}_{n+m-r-1} \\ (s(\vec{x}), t(\vec{x})) & \mapsto & s(\vec{x})f(\vec{x}) + t(\vec{x})g(\vec{x}) \end{array}$$

これは，$f(\vec{x})$ と $g(\vec{x})$ に関する r 次の部分終結式写像であり，$r = 0, \ldots, \min\{n,m\} - 1$ である。ここで，$\mathcal{P}_d$ は全次数が d の多項式全体の集合を表し，$f, g \in \mathbb{Z}[x_1, \ldots, x_\ell]$，$n = \mathrm{tdeg}(f)$，$m = \mathrm{tdeg}(g)$ とする (写像自体は $\mathbb{Z}$ 上に限定されないが，本報告では $\mathbb{Z}$ 上に限定する)。整数係数多項式の近似 GCD では，部分終結式写像が単射でない最大の r に対して，$f(\vec{x})/t(\vec{x})$ と $g(\vec{x})/s(\vec{x})$ は，$f(\vec{x})$ と $g(\vec{x})$ の GCD となる性質を格子算法などで求めることで，近似 GCD や近似余因子の候補を計算する。

例 2 (格子算法による近似 GCD と近似余因子の計算)
次の非常に単純な多項式で，部分集結式写像と格子算法による近似 *GCD* の計算を説明する。

$$\begin{array}{rclcl} f(\vec{x}) & = & 49x_1^2 - 24x_2^2 & = & (7x_1 - 5x_2)(7x_1 + 5x_2) + x_2^2, \\ g(\vec{x}) & = & 50x_1^2 + 70x_1x_2 + 25x_2^2 & = & (7x_1 + 5x_2)(7x_1 + 5x_2) + x_1^2. \end{array}$$

まず，$f(\vec{x})$ と $g(\vec{x})$ の *Sylvester* 行列に単位行列を付与した行列 $Syl_0^E(f,g)$ を作成する。単位行列は，格子算法で求まる短いベクトルの構成要素を知るためのものであり，利用する格子算法のライブラリや実装によっては，この部分は不要である。なお，*Sylvester* 行列部分にはスケーリングが必要であることなど，詳細は参考文献 *[5]* を参照されたい。

$$Syl_0^E(f,g) = \left(\begin{array}{cccccc|cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 49 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & 70 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & 70 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 & 70 & 50 \end{array}\right)$$

上記行列の行ベクトルの張る格子上の短いベクトルを格子算法で求めると，次のようなベクトルが求まる。

$$\left(\begin{array}{cccccc|cccccccccc} 0 & -2 & -3 & 0 & -2 & 3 & 0 & 0 & 0 & -2 & 0 & 0 & 7 & 0 & 12 & 3 \\ 0 & \underline{-5} & \underline{-7} & 0 & \underline{-5} & \underline{+7} & 0 & 0 & 0 & -5 & 0 & 0 & -7 & 0 & -5 & 7 \\ 0 & 7 & 9 & 0 & 6 & -9 & 0 & 0 & 0 & -18 & 0 & 0 & -21 & 0 & 13 & -9 \\ 0 & 10 & 15 & 0 & 10 & -14 & 0 & 0 & 0 & 10 & 0 & 0 & -10 & 0 & 10 & 35 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 49 & 0 & 0 & 70 & 0 & 1 & 0 & 0 \end{array}\right)$$

下線部は近似余因子 *($7x_1 - 5x_2$ と $7x_1 + 5x_2$)* に対応しており，これらと元の $f(\vec{x})$ と $g(\vec{x})$ から次の特殊な行列を構成して，近似 *GCD* を求める *(*近似余因子のため，単純な徐算では *GCD* を求められないため*)*。

$$H(f,g,t,s) = \left(\begin{array}{cccc|cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 49 & 0 & 0 & 25 & 0 & 70 & 50 \\ 0 & 1 & 0 & 0 & 0 & -5 & 0 & 7 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -5 & 0 & 7 & 0 & 0 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -5 & 7 & 0 & 0 & 0 & 0 & 5 & 7 \end{array}\right)$$

再度，行ベクトルの張る格子上の短いベクトルを格子算法で求めると，次のようなベクトルが求まる。

$$H(f,g,t,s) \rightarrow \left(\begin{array}{cccc|cccccccccc} 1 & 0 & \underline{-5} & \underline{-7} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & -5 & 0 & 7 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -5 & 0 & 7 & 0 & 0 & 0 & 5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -5 & 7 & 0 & 0 & 0 & 0 & 5 & 7 \end{array}\right)$$

下線部が近似 *GCD* の係数ベクトルに対応しており，近似 *GCD* として「$7x_1 + 5x_2$」が，誤差 *(*摂動*)* として「1」が求まったことになる。 ◁

3 整数の近似GCDへの拡張

整数の近似 GCD に関しては，Gentry らの論文 [1] に既知の研究が簡潔にサーベイされている。それによれば，Howgrave-Graham の連分数展開や Modular Equation を利用したもの，Lagarias の Simultaneous Diophantine Approximation を利用したもの，Ngyyen と Stern によるサンプルに直行する格子を利用するもの，Ex. Coppersmith による直行関係を複数含む格子を利用したものなどがある。本報告では，Gathen ら [2] が Howgrave-Graham による整数の近似 GCD を整数多項式の近似 GCD に拡張したのとは逆に，著者による整数多項式の近似 GCD を整数の近似 GCD に適用できないかを (まずは) コスト度外視で検討した。

3.1 単純な適用方法

整数係数多項式と整数を次の写像により，1 対 1 対応を行うことで，先の算法を適用することを考える。

$$\begin{array}{rccc} \phi: & \mathbb{Z} & \rightarrow & \mathbb{Z}[x] \\ & \sum_{i=0}^{n} a_i 10^i \ (a_i \in \{0,1,\ldots,9\}) & \mapsto & \sum_{i=0}^{n} a_i x^i \end{array}$$

この対応は 10 進法表記をそのまま多項式にするものであり，次のような関係になっている。

$$123456 = 1 \times 10^5 + 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 \Rightarrow x^5 + 2x^4 + 3x^3 + 4x^2 + 5x + 6$$
$$450608 = 4 \times 10^5 + 5 \times 10^4 + 6 \times 10^2 + 8 \times 10^0 \Rightarrow 4x^5 + 5x^4 + 6x^2 + 8$$

しかしながら，この写像は準同型でなく，特に加減乗算時の桁上がりと桁下がりが実現できないため，整数係数多項式の近似 GCD 算法をそのまま整数の近似 GCD 計算に使用することはできない。

3.2 桁上がりと桁下がりを実現する格子による方法

格子の近似最短ベクトルを計算する上で，先の写像が線形写像となるように，次の基底を導入する。

$$\vec{u}_i = (0,\ldots,0,-1,10,0,\ldots,0) \quad (\text{第 } i \text{ 成分が } 10 \text{ で，第 } (i-1) \text{ 成分が } -1)$$

この基底 $\vec{u}$ を加えると「桁下がり」(第 $(i-1)$ 成分から桁を借り，第 i 成分が 10 増える) が実現でき，$\vec{u}$ を減じると「桁上がり」(第 i 成分から 10 を引き，第 $(i-1)$ 成分が 1 増える) が実現できる。そのため，$\vec{u}$ をシフトして必要な分だけを格子に加えることで，格子の基底ベクトル間の加減算時の桁上がりと桁下がりが自然に行えるようになる。この方法は，単純なので何かに掲載されているかもしれないが少なくとも著者は見たことがない。

例 3 ($\vec{u}$ による整数の GCD 計算 (厳密な場合))
実際に，次の関係を用いて，多項式の部分集結式写像による整数の *GCD* を求めてみる。

$$\begin{aligned} c_1 &= 325 \times 78 = 25350 = 2x^4 + 5x^3 + 3x^2 + 5x, \\ c_2 &= 432 \times 78 = 33696 = 3x^4 + 3x^3 + 6x^2 + 9x + 6. \end{aligned}$$

Sylvester 行列部分 *(*スケーリングされていることに注意*)* に単位行列を付与した行列は次の通り。

$$\left(\begin{array}{cccccc|ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & -60 & -90 & -60 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & -60 & -90 & -60 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & -60 & -90 & -60 \\ 0 & 0 & 0 & 1 & 0 & 0 & 20 & 50 & 30 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 20 & 50 & 30 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 20 & 50 & 30 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 \end{array}\right)$$

この行列の行ベクトルが張る格子に対して，格子算法で短いベクトルを計算すると，次のベクトルが得られる。下線部のところに，余因子 *(*325 と 432*)* が検出されていることがわかる。桁が下がりと桁上がりが実現されているため，325 の表現が一意でないことに注意されたい。

$$\left(\begin{array}{cccccc|ccccccc} \underline{-3} & \underline{-2} & \underline{-5} & \underline{-4} & \underline{-3} & \underline{-2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \underline{+3} & \underline{+3} & \underline{-5} & \underline{+4} & \underline{+3} & \underline{+2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 5 & 1 & 1 & -1 & 0 & -10 & 0 & -10 & 0 & -10 & 0 \\ -2 & 1 & 0 & -2 & -5 & 1 & 0 & 10 & -10 & 0 & 10 & -10 & 0 \\ 0 & -5 & 0 & -1 & 0 & -2 & -10 & 10 & 0 & -10 & 10 & 0 & 0 \\ 2 & 0 & 0 & 2 & 3 & -4 & -10 & 0 & -10 & 0 & -10 & 0 & 0 \\ -1 & 1 & 0 & -2 & 4 & 1 & -10 & 0 & 0 & 20 & 0 & -10 & 0 \\ 1 & 1 & 0 & 1 & 5 & -3 & 0 & 0 & 20 & 0 & -10 & -10 & 0 \\ -1 & -4 & 2 & -2 & 2 & -3 & 0 & 0 & 10 & 10 & 0 & 0 & -20 \end{array}\right)$$

◁

例 4 ($\vec{u}$ による整数の近似 GCD 計算)
同様に，次の整数に対して，多項式の部分集結式写像による整数の近似 *GCD* を求めてみる。

$$\begin{aligned} c_1 &= 325 \times 78 + 2 = 25352 = 2x^4 + 5x^3 + 3x^2 + 5x + 2, \\ c_2 &= 432 \times 78 - 1 = 33695 = 3x^4 + 3x^3 + 6x^2 + 9x + 5. \end{aligned}$$

Sylvester 行列部分 (スケーリングされていることに注意) に単位行列を付与した行列は次の通り。

$$\left(\begin{array}{cccccc|ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & -60 & -90 & -50 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & -60 & -90 & -50 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -30 & -30 & -60 & -90 & -50 \\ 0 & 0 & 0 & 1 & 0 & 0 & 20 & 50 & 30 & 50 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 20 & 50 & 30 & 50 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 20 & 50 & 30 & 50 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 \end{array}\right)$$

この行列の行ベクトルが張る格子に対して，格子算法で短いベクトルを計算すると，次のベクトルが得られる。残念ながら，近似 *GCD* に対応する近似余因子の検出に成功している状態とは言えない。

$$\left(\begin{array}{cccccc|ccccccc} 0 & 0 & 0 & -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 4 & 2 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & -10 & -10 & 0 \\ -2 & -1 & 2 & -3 & 2 & 0 & 0 & -10 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 1 & -3 & 0 & 0 & 10 & -10 & 0 & -10 & -10 \\ -6 & -4 & -1 & -8 & -5 & 2 & 0 & 10 & 0 & 0 & 0 & 0 & -10 \\ 0 & 2 & -3 & 1 & -4 & 2 & 10 & 0 & 0 & -10 & 0 & 10 & -10 \\ -1 & -2 & 1 & -2 & 0 & 2 & -10 & 0 & -10 & 0 & 0 & 10 & -10 \\ -4 & -1 & 4 & -5 & -4 & 0 & 0 & 0 & -10 & 0 & 10 & -10 & 0 \\ 6 & -3 & 3 & 8 & -4 & 2 & 0 & 0 & 10 & 10 & 0 & -10 & -10 \end{array}\right)$$

◁

これらの計算例で，期待した結果にならない原因は，多項式の場合と同じである。整数係数多項式の近似 GCD 算法においても，各単項式における摂動が大きい場合は顕著であるが，求まって欲しい近似 GCD が得られることは少ない。この原因は，部分集結式写像の核 ($sc_1 + tc_2 = 0$) の限界であり，近似余因子は $sc_1 + tc_2 \approx 0$ を満たすとは限らないため，格子算法による近似最短ベクトルが近似余因子に対応しないことが発生する。これは，$s(c_1 + \varepsilon_1) + t(c_2 + \varepsilon_2) = s\varepsilon_1 + t\varepsilon_2$ のためである。

3.3 誤差を考慮した重み付け

まず，整数係数多項式の近似 GCD と同じ改善策としては，近似 GCD のサイズ (近似余因子のサイズ) が既知の場合に，部分集結式写像の次数 r をなるべく大きく取る (行列サイズは小さくなる) ことが挙げられる。整数の近似 GCD で構成する格子に特有な改善策としては，$s\varepsilon_1 + t\varepsilon_2$ の上限を推測可能 (例えば，セキュリティパラメータなどから推測可能) であれば，格子の基底ベクトルの要素に重みを付け，下位の桁を比較的無視するようにすれば良い[1)]。

[1)] これは，構成する格子サイズを大きくすれば多項式にも適用可能と思われるが，本報告の目的でないので取り扱わない。

例 5 ($\vec{u}$ による整数の近似 GCD 計算 (重み付けあり))
この改善策で次の整数の近似 *GCD* を，多項式の部分集結式写像で求めてみる。

$$\begin{aligned} c_1 &= 325 \times 78 + 2 = 25352 = 2x^4 + 5x^3 + 3x^2 + 5x + 2, \\ c_2 &= 432 \times 78 - 1 = 33695 = 3x^4 + 3x^3 + 6x^2 + 9x + 5. \end{aligned}$$

Sylvester 行列部分に単位行列を付与した行列は次の通り。先ほどの例とは異なり，下位の桁 *(*右から *4* 列分*)* に比べて，上位の桁 *(*左から *3* 列分*)* のスケーリングを大きくしていることに注意されたい。

$$\left(\begin{array}{cccccc|ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & -300 & -300 & -600 & -90 & -50 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -300 & -300 & -60 & -90 & -50 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -300 & -30 & -60 & -90 & -50 \\ 0 & 0 & 0 & 1 & 0 & 0 & 200 & 500 & 300 & 50 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 200 & 500 & 30 & 50 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 200 & 50 & 30 & 50 & 20 \\ 0 & 0 & 0 & 0 & 0 & 0 & -100 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -100 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10 & 100 \end{array}\right)$$

この行列の行ベクトルが張る格子に対して，格子算法で短いベクトルを計算すると，次のベクトルが得られる。今回は，下線部のところに近似 *GCD* に対応する近似余因子を検出することに成功している。

$$\left(\begin{array}{cccccc|ccccccc} 0 & 0 & 0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 4 & 2 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & -10 & -10 & 0 \\ 6 & -3 & -2 & 7 & 5 & 5 & 0 & 0 & 0 & 20 & 0 & 0 & 0 \\ -8 & -5 & -4 & -11 & -3 & -5 & 0 & 0 & 0 & 10 & 0 & 10 & 0 \\ -2 & 3 & 3 & -2 & -2 & -2 & 0 & 0 & 0 & -10 & -10 & 20 & 10 \\ \underline{-3} & \underline{-1} & \underline{-3} & \underline{-4} & \underline{-2} & \underline{+4} & 0 & 0 & 0 & 0 & 10 & 0 & 30 \\ 0 & 0 & -5 & 0 & -1 & 3 & 0 & 0 & -100 & 10 & 0 & 10 & 10 \\ 6 & 4 & 1 & 8 & 5 & -2 & 0 & -100 & 0 & 0 & 0 & 0 & 10 \\ 0 & 2 & -3 & 1 & -4 & 2 & 100 & 0 & 0 & -10 & 0 & 10 & -10 \end{array}\right)$$

この近似余因子の係数ベクトルを，桁上がりと桁下がりを考慮して戻すと，次の 313 と 416 が得られる。

$$(-3 \quad -1 \quad -3) \Rightarrow -313, \quad (-4 \quad -2 \quad +4) \Rightarrow -416$$

この結果は，予期した 325×78 と 432×78 と異なるが，摂動 *(*誤差*)* を計算すると予期した値 *(2)* よりも小さいことがわかる。つまり，整数の近似 *GCD* の計算に成功していると言える。

$$|25352 - 313 \times 81| = 1, \quad |33695 - 416 \times 81| = 1$$

◁

参 考 文 献

[1] M. Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, Fully Homomorphic Encryption over the Integers. EUROCRYPT 2010, LNCS 6110, 2010, 24–43.

[2] J. Gathen, M. Mignotte and I. E. Shparlinski, Approximate polynomial GCD: Small degree and small height perturbations. J. Symbolic Computation, Vol. 45, No. 8, 2010, 879–886.

[3] C. Gentry, Computing Arbitrary Functions of Encrypted Data. Communications of the ACM, Vol. 53, No. 3, 2010, 97–105.

[4] C. Gentry, A fully homomorphic encryption scheme. Ph.D Thesis, Stanford University, 2009.

[5] K. Nagasaka, Approximate Polynomial GCD over Integers. submitted (2009), extended abstract published in ACM Communications in Computer Algebra. Vol. 42(3). 2008. 124–126.